现代程序设计第12周作业

谢奕飞 20377077

代码

WriteHead函数

```python
def WriteHead(head,dir_path):
    '''
    写入表头
    '''
    if not os.path.exists(dir_path):
        os.mkdir(dir_path)
    with open(dir_path+'/data.csv','w',encoding='utf8',newline='') as f:
        writer=csv.writer(f)
        writer.writerow(head)
```

GetPage函数

```python
def GetPage(url,headers):
    '''
    获取页数
    '''
    response=requests.get(url=url,headers=headers)
    soup=BeautifulSoup(response.text,'lxml')
    return int(soup.select('a[class="zpgi"]')[-1].get_text())
```

Producer函数

```python
def Producer(q:Queue,url,headers):
    '''
    生产者
    '''
    response=requests.get(url=url,headers=headers)
    soup=BeautifulSoup(response.text,'lxml')
    soup_list=soup.select('a[class="tit f-thide s-fc0"]')
    href_list=[]
    for s in soup_list:
        href_list.append('https://music.163.com'+str(s['href']))
    q.put(href_list)
```

Consumer

```python
def Consumer(q:Queue,headers,dir_path):
    '''
    消费者
    '''
    head=['id','title','image','author_id','author','description','count','play','add','share','comment']
    urls=q.get()
    if urls!=None:
        for url in urls:
            response=requests.get(url=url,headers=headers)
            soup=BeautifulSoup(response.text,'lxml')
            id=url.split('id=')[-1]
            title=soup.select('.tit')[0].get_text()[1:]
            image_url=soup.select('img[class="j-img"]')[0]['data-src']
            image=DownloadImage(image_url,dir_path+'/images')
            author_id=soup.select('a[class="s-fc7"]')[0]['href'].split('id=')[-1]
            author=soup.select('a[class="s-fc7"]')[0].get_text()
            description=soup.select('p')[1].get_text()
            count=soup.select('span[id="playlist-track-count"]')[0].get_text()
            play=soup.select('strong[id="play-count"]')[0].get_text()
            add=soup.select('a[class="u-btni u-btni-fav"]')[0]['data-count']
            share=soup.select('a[class="u-btni u-btni-share"]')[0]['data-count']
            comment=soup.select('span[id="cnt_comment_count"]')[0].get_text()
            with open(dir_path+'/data.csv','a',encoding='utf8',newline='') as f:
                writer=csv.writer(f)
                writer.writerow([id,title,image,author_id,author,description,count,play,add,share,comment])
```

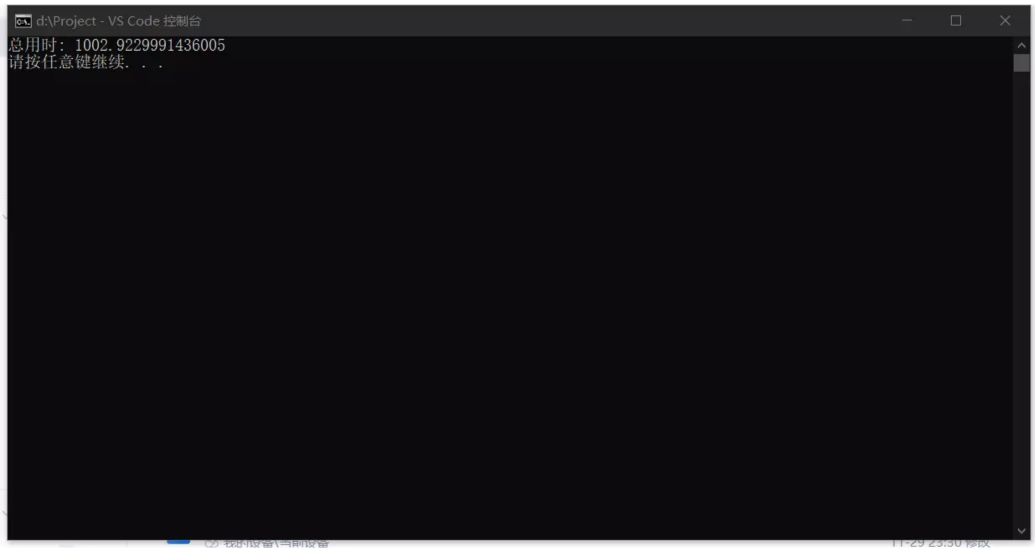
DownloadImage函数

```
def DownloadImage(url,dir_path):
    '''
    下载图片
    '''
    if not os.path.exists(dir_path):
        os.mkdir(dir_path)
    name=url.split('/')[-1]
    img_path=dir_path+'/'+name
    try:
        urllib.request.urlretrieve(url,filename=img_path)
        urllib.request.urlcleanup()
    except:
        return 'error'
    return img_path
```

main函数

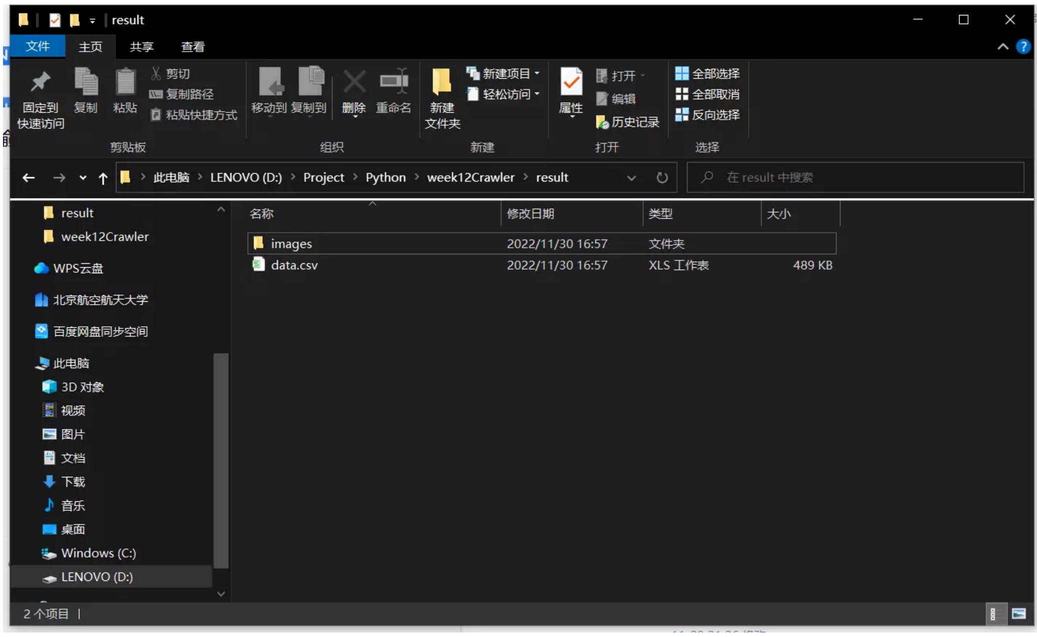
```python
if __name__=='__main__':
    t_start=time.time()
    url='https://music.163.com/discover/playlist/?order=hot&cat=%E6%B0%91%E8%B0%A3&limit=35&offset=0'
    headers={'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.35'}
    dir_path='D:/Project/Python/week12Crawler/result'
    head=['id','title','image','author_id','author','description','count','play','add','share','comment']
    WriteHead(head,dir_path)#写入表头
    N=GetPage(url,headers)#获取页数
    urls=[]
    for i in range(N):
        urls.append(f'https://music.163.com/discover/playlist/?order=hot&cat=%E6%B0%91%E8%B0%A3&limit=35&offset={i*35}')
    q=Queue()
    #Producer(q,url,headers)
    #Consumer(q,headers,dir_path)

    #爬
    plist,clist=[],[]
    for url in urls:
        p=Thread(target=Producer,args=(q,url,headers,))
        plist.append(p)
    for i in range(N):
        c=Thread(target=Consumer,args=(q,headers,dir_path,))
        clist.append(c)

    for p in plist:
        p.start()
    for c in clist:
        c.start()
    for p in plist:
        p.join()
    for c in clist:
        q.put(None)#主进程发信号结束，但要给每一个consumer准备
    for c in clist:
        c.join()
    t_finish=time.time()
    print('总用时: {}'.format(t_finish-t_start))
```

运行结果



```
总用时: 1002.9229991436005
请按任意键继续. . .
```

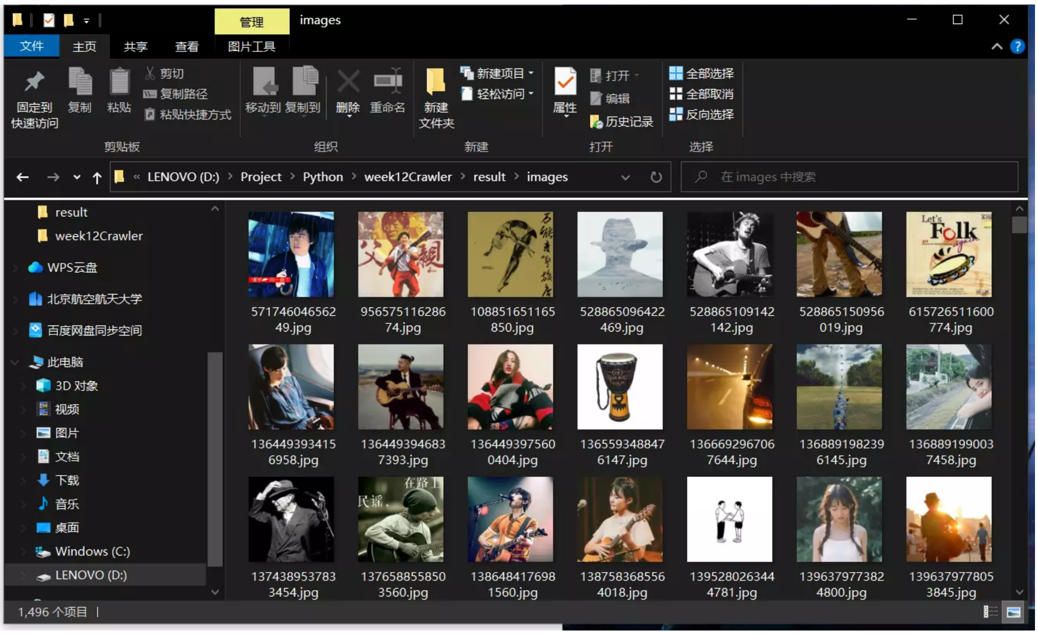
用时1002.92s



此电脑 › LENOVO (D:) › Project › Python › week12Crawler › result

名称	修改日期	类型	大小
images	2022/11/30 16:57	文件夹	
data.csv	2022/11/30 16:57	XLS 工作表	489 KB

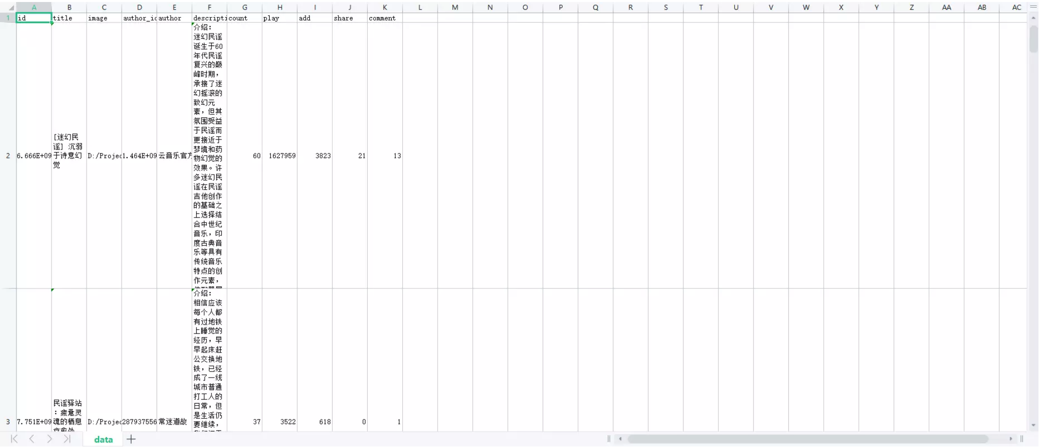
2 个项目



« LENOVO (D:) › Project › Python › week12Crawler › result › images

57174604656249.jpg
95657511628674.jpg
108851651165850.jpg
528865096422469.jpg
528865109142142.jpg
528865150956019.jpg
615726511600774.jpg
1364493934156958.jpg
1364493946837393.jpg
1364493975600404.jpg
1365593488476147.jpg
1366692967067644.jpg
1368891982396145.jpg
1368891990037458.jpg
1374389537833454.jpg
1376588558503560.jpg
1386484176981560.jpg
1387583685564018.jpg
1395280263444781.jpg
1396379773824800.jpg
1396379778053845.jpg

1,496 个项目



id	title	image	author_ic	author	descripti	count	play	add	share	comment
6.666E+09	[迷幻民谣] 沉溺于诗意幻觉	D:/Projec	1.464E+09	云音乐官方	介绍：迷幻民谣诞生于60年代民谣复兴的巅峰时期，承接了迷幻摇滚的致幻元素，但其氛围受益于民谣而更接近于梦境和药物幻觉的效果。许多迷幻民谣在民谣吉他创作的基础之上选择结合中世纪音乐，印度古典音乐等具有传统音乐特点的创作元素，	60	1627959	3823	21	13
7.751E+09	民谣驿站：疲惫灵魂的栖息	D:/Projec	287937556	常迷道故	介绍：相信应该每个人都有过地铁上睡觉的经历，早早起床赶公交换地铁，已经成了一线城市普通打工人的日常，但是生活仍要继续，	37	3522	618	0	1

附录–完整代码

```
import requests
from bs4 import BeautifulSoup
import time
from queue import Queue
from threading import Thread
import csv
import os
import urllib.request

def WriteHead(head,dir_path):
    '''
    写入表头
    '''
    if not os.path.exists(dir_path):
        os.mkdir(dir_path)
    with open(dir_path+'/data.csv','w',encoding='utf8',newline='') as f:
        writer=csv.writer(f)
        writer.writerow(head)

def GetPage(url,headers):
    '''
    获取页数
    '''
    response=requests.get(url=url,headers=headers)
    soup=BeautifulSoup(response.text,'lxml')
    return int(soup.select('a[class="zpgi"]')[-1].get_text())

def Producer(q:Queue,url,headers):
    '''
    生产者
    '''
    response=requests.get(url=url,headers=headers)
    soup=BeautifulSoup(response.text,'lxml')
    soup_list=soup.select('a[class="tit f-thide s-fc0"]')
    href_list=[]
    for s in soup_list:
        href_list.append('https://music.163.com'+str(s['href']))
    q.put(href_list)

def Consumer(q:Queue,headers,dir_path):
    '''
    消费者
    '''
    head=['id','title','image','author_id','author','description','count','play','add','share','comment']
    urls=q.get()
    if urls!=None:
        for url in urls:
            response=requests.get(url=url,headers=headers)
            soup=BeautifulSoup(response.text,'lxml')
            id=url.split('id=')[-1]
            title=soup.select('.tit')[0].get_text()[1:]
            image_url=soup.select('img[class="j-img"]')[0]['data-src']
            image=DownloadImage(image_url,dir_path+'/images')
            author_id=soup.select('a[class="s-fc7"]')[0]['href'].split('id=')[-1]
            author=soup.select('a[class="s-fc7"]')[0].get_text()
            description=soup.select('p')[1].get_text()
            count=soup.select('span[id="playlist-track-count"]')[0].get_text()
            play=soup.select('strong[id="play-count"]')[0].get_text()
            add=soup.select('a[class="u-btni u-btni-fav"]')[0]['data-count']
            share=soup.select('a[class="u-btni u-btni-share"]')[0]['data-count']
            comment=soup.select('span[id="cnt_comment_count"]')[0].get_text()
            with open(dir_path+'/data.csv','a',encoding='utf8',newline='') as f:
                writer=csv.writer(f)
                writer.writerow([id,title,image,author_id,author,description,count,play,add,share,comment])

def DownloadImage(url,dir_path):
    '''
    下载图片
    '''
    if not os.path.exists(dir_path):
        os.mkdir(dir_path)
    name=url.split('/')[-1]
    img_path=dir_path+'/'+name
    try:
        urllib.request.urlretrieve(url,filename=img_path)
        urllib.request.urlcleanup()
    except:
        return 'error'
    return img_path

```

```python
if __name__=='__main__':
    t_start=time.time()
    url='https://music.163.com/discover/playlist/?order=hot&cat=%E6%B0%91%E8%B0%A3&limit=35&offset=0'
    headers={'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.35'}
    dir_path='D:/Project/Python/week12Crawler/result'
    head=['id','title','image','author_id','author','description','count','play','add','share','comment']
    WriteHead(head,dir_path)#写入表头
    N=GetPage(url,headers)#获取页数
    urls=[]
    for i in range(N):
        urls.append(f'https://music.163.com/discover/playlist/?order=hot&cat=%E6%B0%91%E8%B0%A3&limit=35&offset={i*35}')
    q=Queue()
    #Producer(q,url,headers)
    #Consumer(q,headers,dir_path)

    #爬
    plist,clist=[],[]
    for url in urls:
        p=Thread(target=Producer,args=(q,url,headers,))
        plist.append(p)
    for i in range(N):
        c=Thread(target=Consumer,args=(q,headers,dir_path,))
        clist.append(c)

    for p in plist:
        p.start()
    for c in clist:
        c.start()
    for p in plist:
        p.join()
    for c in clist:
        q.put(None)#主进程发信号结束，但要给每一个consumer准备
    for c in clist:
        c.join()
    t_finish=time.time()
    print('总用时：{}'.format(t_finish-t_start))
```